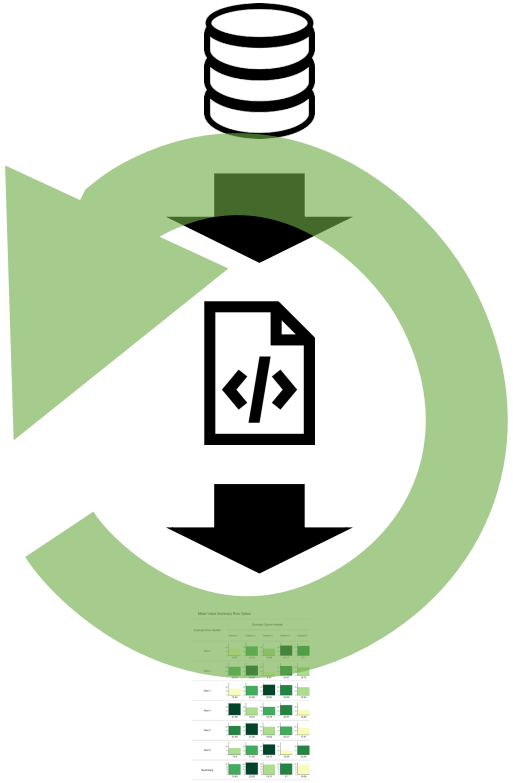
Tips & tricks for making your own R package:

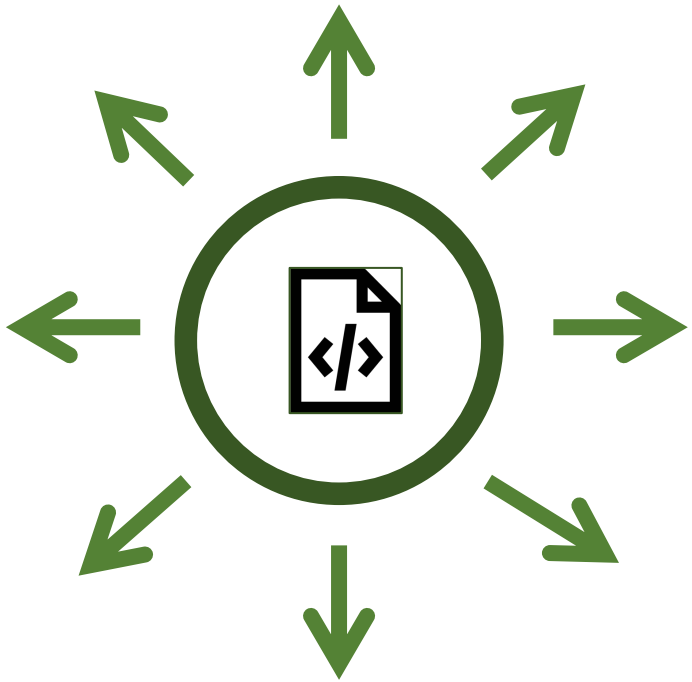
Amanda Hart

Tidal Exchange
4/7/22

Why build your own R package?

1. Organization
2. Reproducibility
3. Distribution
4. Novel tool





Where to start:

R Packages you will need installed

- roxygen2
- devtools

5 Easy steps to build an R package

Where to start:

R Packages you will need installed

- roxygen2
- devtools

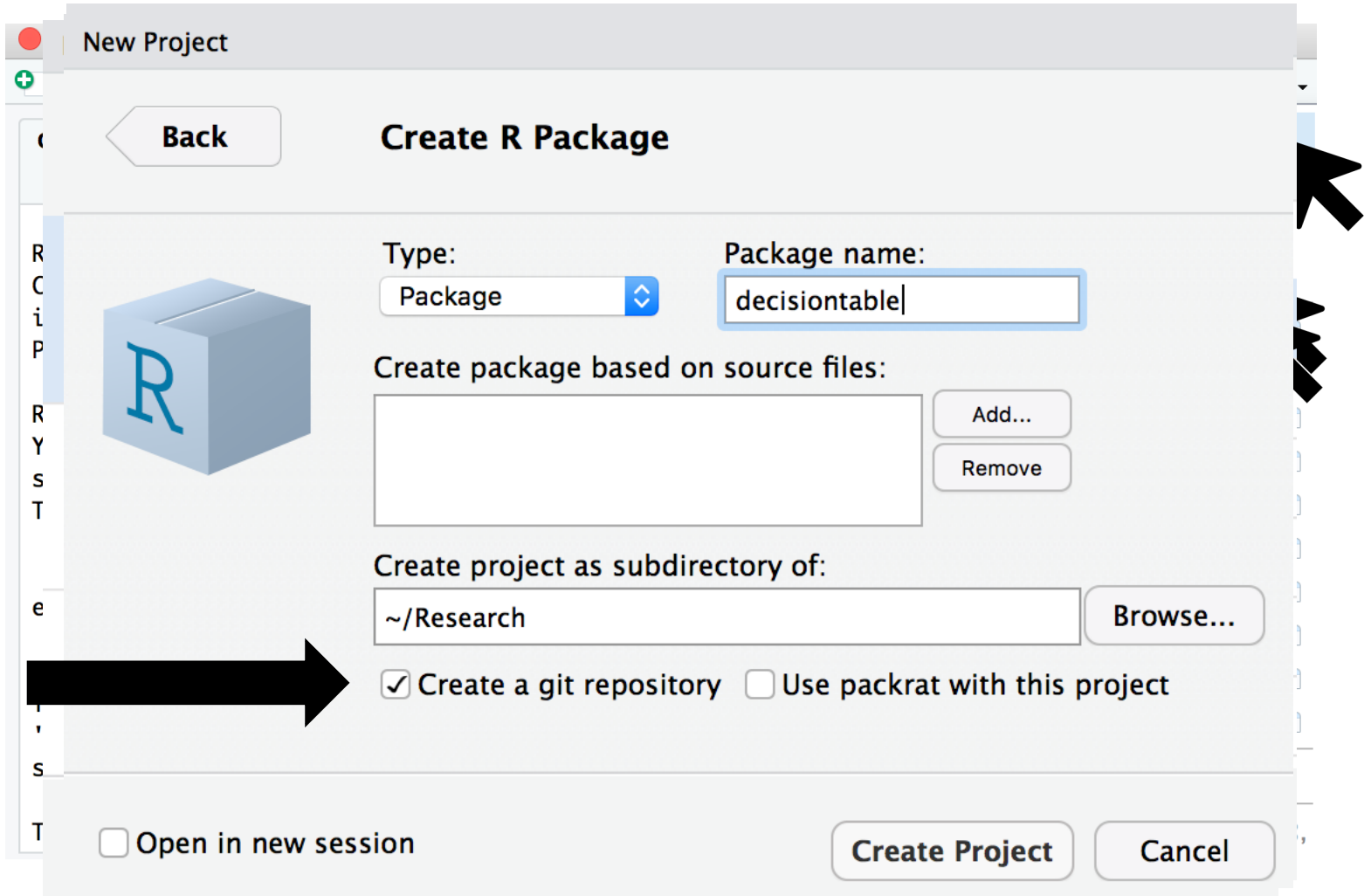
decisiontable R package

	Column 1	Column 2	Column 3	Column 4	Column 5
Row 1	16.86773	22.43715	16.893797	24.10611	23.09913
Row 2	20.91822	23.69162	8.926501	22.96951	19.71936
Row 3	15.82186	22.87891	25.624655	24.59489	19.22102
Row 4	27.9764	18.47306	19.775332	23.91068	12.64624
Row 5	21.64754	27.55891	19.919049	20.37282	17.60925
Row 6	15.89766	21.94922	24.719181	10.05324	22.08971

devtools::install_github("ahart1/decisiontable")

5 Easy steps to build an R package

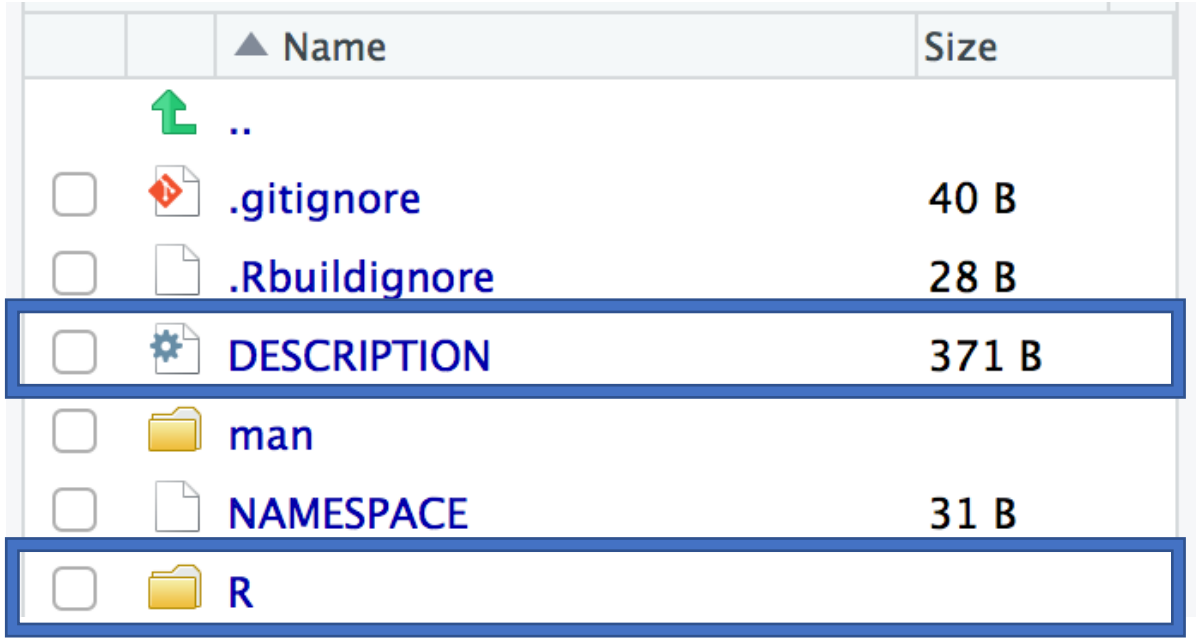
1) Set up your package directory



5 Easy steps to build an R package

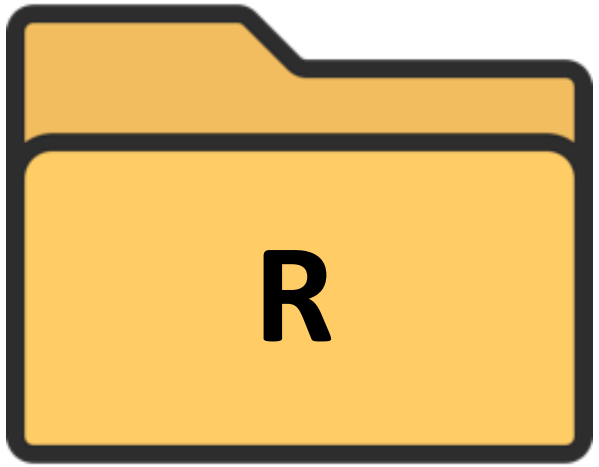
1) Set up your package directory

4 main files



5 Easy steps to build an R package

2) Populate the R folder



GeneralDecisionTableFunction.R

Source on Save | Run | Source

```
131  ######################################################################
132  # Plot graphics
133  ######################################################################
134  # Create png: filename, width, height, resolution can all be adjusted
135  png(filename = paste(OutputDirectory, paste(OutputFileName, ".png", sep=""), se
136
137  # Set up correct graphic format (number of rows/columns, layout matrix) for giv
138  graphicFormat <- NULL # Start with empty format object
139  if(ncol(data) == 1){
140
141    if(SummaryRowOption == "Off"){
142      graphicFormat$graphicNrow <- 6+nrow(data)*2
143      loopnums <- seq(from=11, to=11+nrow(data)*3, by=3)
144    } else{
145      graphicFormat$graphicNrow <- 6+nrow(data)*2 + 2 # Add 2 rows for summary ro
146      loopnums <- seq(from=11, to=11+(nrow(data)+1)*3, by=3) # Add 2 rows for sum
147    }
148    graphicFormat$graphicNcol <- 5
149    graphicLayout <- c(1, 2, 3, 4, 5, rep(6, ncol(data)+3), 5, 7, rep(8, ncol(dat
150    for(inum in loopnums[-length(loopnums)]){
151      graphicLayout <- c(graphicLayout, inum+1, rep(inum+2, ncol(data)+2), 5)
152      graphicLayout <- c(graphicLayout, rep(inum+3, ncol(data)+3), 5)
153    }
154    graphicFormat$graphicLayout <- graphicLayout
155
156  } else if(ncol(data) == 2){
157
158    if(SummaryRowOption == "Off"){
159      graphicFormat$graphicNrow <- 6+nrow(data)*2
160      loopnums <- seq(from=12, to=12+nrow(data)*4, by=4)
161    } else{
162      graphicFormat$graphicNrow <- 6+nrow(data)*2+2 # Add 2 rows for summary row
163      loopnums <- seq(from=12, to=12+(nrow(data)+1)*4, by=4) # Add 2 rows for sum
164    }
```

1:1 (Top Level) R Script

5 Easy steps to build an R package

3) Populate the DESCRIPTION file

DESCRIPTION

```
1  Package: decisiontable
2  Type: Package
3  Title: What the Package Does (Title Case)
4  Version: 0.1.0
5  Author: Who wrote it
6  Maintainer: The package maintainer <yourself@somewhere.net>
7  Description: More about what it does (maybe more than one line)
8      Use four spaces when indenting paragraphs within the Description.
9  License: What license is it under?
10 Encoding: UTF-8
11 LazyData: true
12
```

5 Easy steps to build an R package

3) Populate the DESCRIPTION file

```
Imports:  R (>= 3.4.0),
          ggplot2,
          ggplotify,
          grid,
          gridExtra,
          png,
          raster,
          rasterVis,
          rgdal,
          tidyverse,
          cowplot,
          magick
```

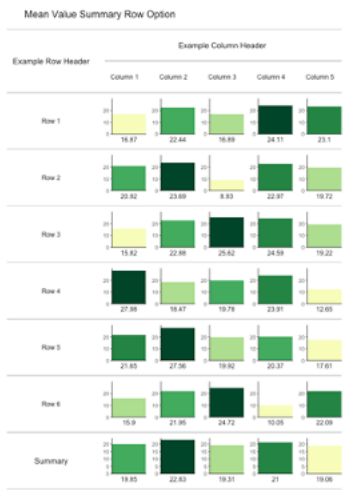
R version

Plotting

Saving final image

Dealing with rasters

	Column 1	Column 2	Column 3	Column 4	Column 5
Row 1	16.86773	22.43715	16.893797	24.10611	23.09913
Row 2	20.91822	23.69162	8.926501	22.96951	19.71936
Row 3	15.82186	22.87891	25.624655	24.59489	19.22102
Row 4	27.9764	18.47306	19.775332	23.91068	12.64624
Row 5	21.64754	27.55891	19.919049	20.37282	17.60925
Row 6	15.89766	21.94922	24.719181	10.05324	22.08971



Trick: Search your R script for library() function calls, if you needed to load a package for your original script to run then you will need that package loaded as a dependency to your new package

5 Easy steps to build an R package

4) Document everything!!!

Use the **roxygen2** R package to document your functions using the #’ comment and @ tag notation

Each function needs:

#’ @title A title for the documentation page

#’ @description A description of your function

#’ that may span multiple lines.

#’ @param parameterName Followed by a description of this parameter input to the function, include description of default setting

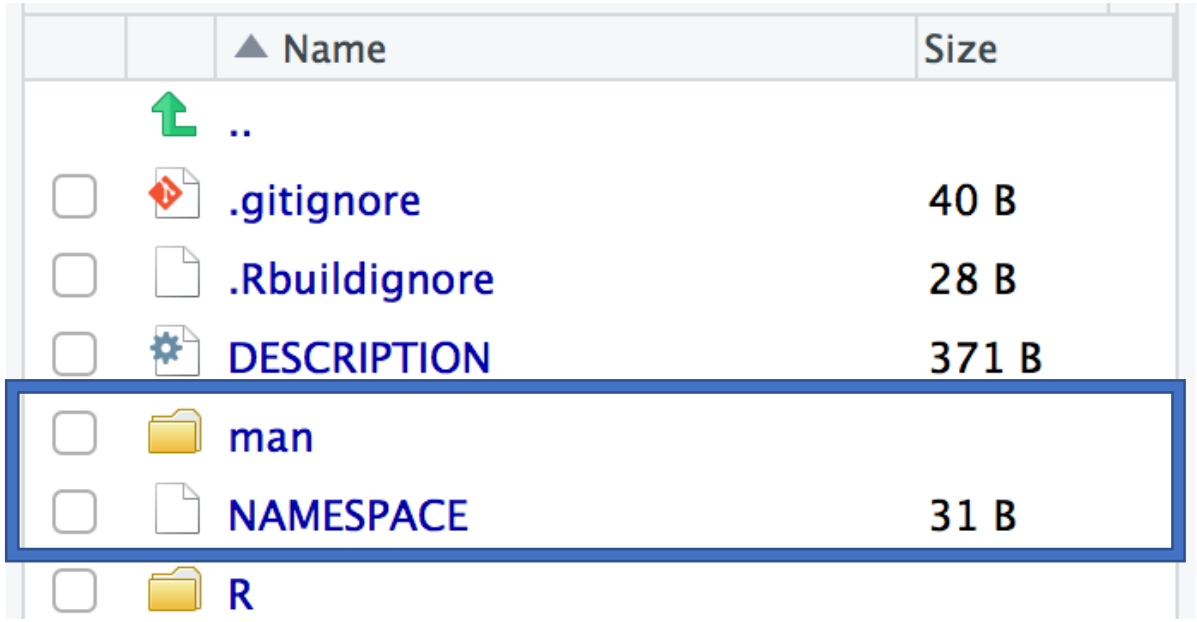
#’ @return Describing what the function returns (e.g. a number, a table, an image file)

#’ @examples

5 Easy steps to build an R package

4) Document everything!!!

4 main files

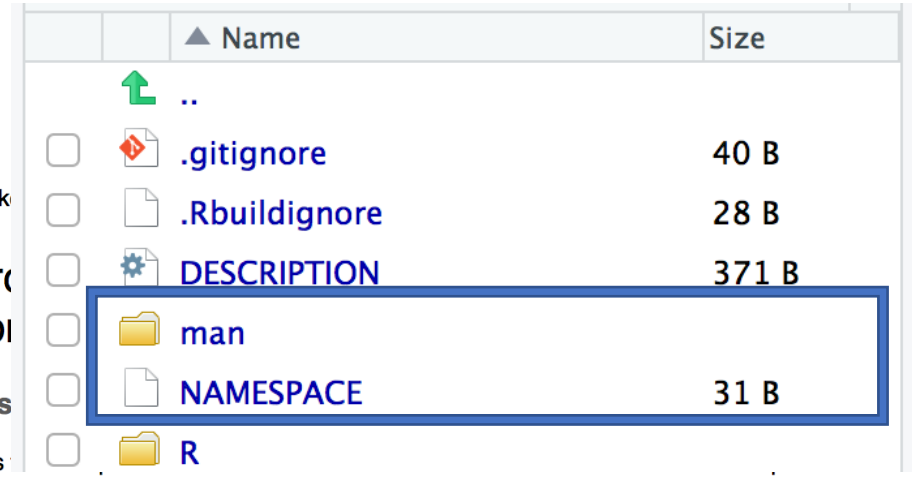


5 Easy steps to build an R package

5) Build and test your package

- To compile and format your documentation run: **document()**
  - Check that the **man** folder has .Rd files generated by roxygen2
  - Check that the **NAMESPACE** file has been populated by roxygen2 (if not did you remember to add #' @export to your documentation?)

- Load the package locally using: **library()**
  - You can now view the formatted help documentation by calling **?**functionname
  - Try restarting Rstudio if documentation is not visible or returns an error



matrix or the specified dimensions if data is provided as vector. Labels, figure layout and dimensions, and coloring scheme, can be customized. Confidence intervals may be specified using the upper bounds provided in Data_UpperCI and lower bounds provided in Data_LowerCI data inputs (confidence intervals are not automatically generated). Up to three icons may be printed next to the title, and a summary information can also be automatically calculated and appended as the last row of the decision table.

Usage

```
makeDecisionTable(
  data,
  rownames = NULL,
  colnames = NULL,
  OutputDirectory = getwd(),
  OutputFileName = "DecisionTable",
  GraphicTitle = "Title",
```

5 Easy steps to build an R package

5) Build and test your package

Restart R to run final tests to ensure others can use it:

- Try installing your package
  - E.g. **install("decisiontable")** OR **devtools::install_github("ahart1/decisiontable")**
- Load your package again and run the examples
  - If they don't work, check that you included:
    - All dependent packages
    - All example datasets

“Advanced” features & Next steps

Share your package

- Add your package to CRAN
  - http://r-pkgs.had.co.nz/release.html
- Host your package on GitHub
  - https://kbroman.org/pkg_primer/pages/github.html

“Advanced” features & Next steps

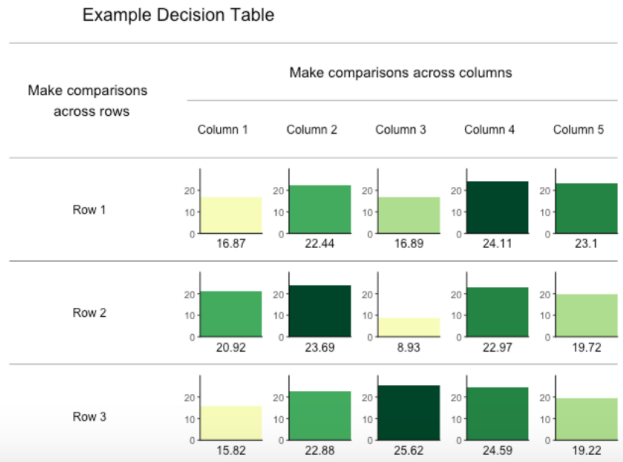
Expand your documentation

- Add a README file: by running **use_readme_rmd()** (in devtools package) in your R package directory
  - https://usethis.r-lib.org/reference/use_readme_rmd.html
- Add a vignette
  - http://r-pkgs.had.co.nz/vignettes.html

README.md

decisiontable

The goal of decisiontable is to combine text-based tables with graphical elements to produce flexible decision table visualizations that convey trade-offs and are easily reproduced. The package leverages bar plots, coloring options, layout choices, and optional summary information to highlight and display trade-offs.



Installation

You can install the development version of decisiontable from GitHub with:

```
# install.packages("devtools")
devtools::install_github("ahart1/decisiontable")
```

“Advanced” features & Next steps

Extra features

- R datasets
- Explore roxygen2 documentation features
  - E.g. use @export to indicate functions that should be visible to users of the package vs. those that should stay behind the scenes
  - Explore the NAMESPACE file: http://r-pkgs.had.co.nz/namespace.html
  - Use @inheritParams to inherit parameter documentation from another function

Helpful tutorials

- R Packages book: http://r-pkgs.had.co.nz/
- Simple R package example: https://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/